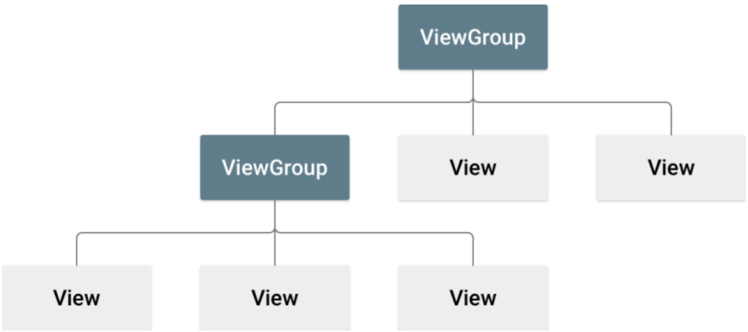
Aula 3 - Pizza examples (UI interface)

Summary:

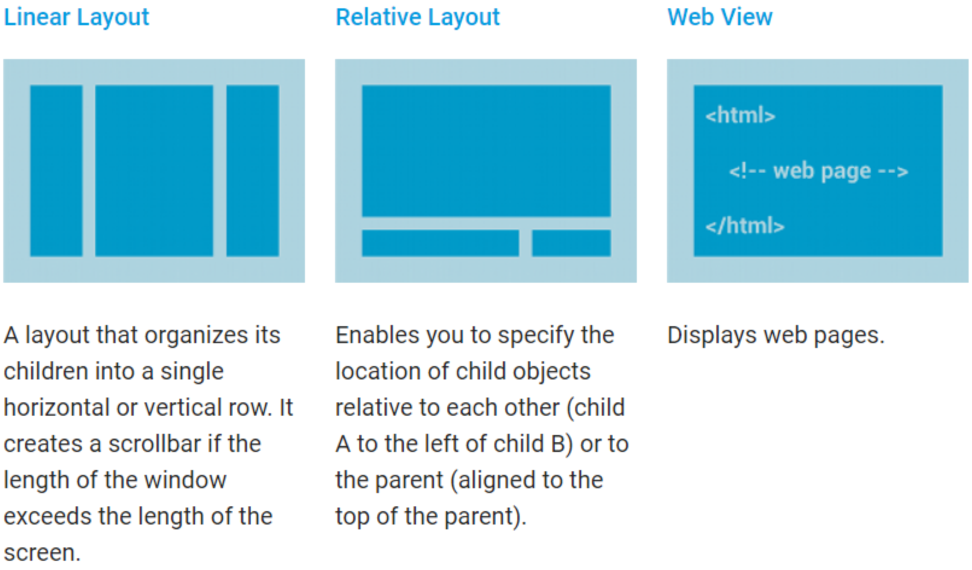
- Layouts
- List Activity
- GridViews
- Portrait and Landscape Layout
- Theme Editor

3.1. Layouts

The layout organizes the structure of the app. There are built using a hierarchy of Views (a basic UI element user can see and interact with) and ViewGroups (invisible container that defines the layout for views).



Common layouts are available to organize the UI:



Linear Layout

A layout that organizes its children into a single horizontal or vertical row. It creates a scrollbar if the length of the window exceeds the length of the screen.

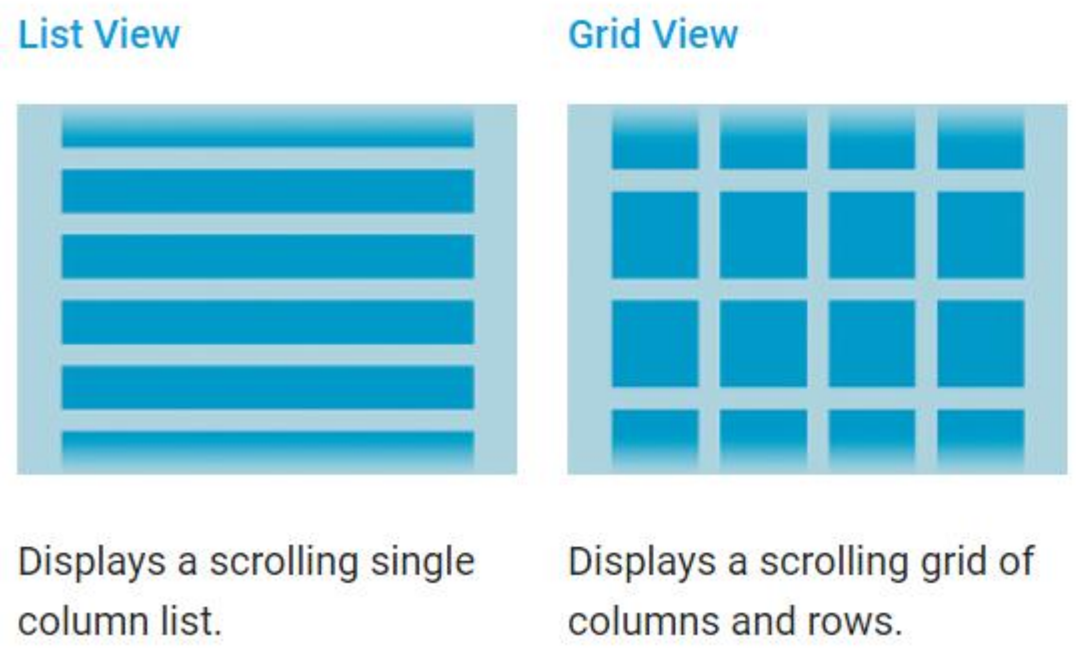
Relative Layout

Enables you to specify the location of child objects relative to each other (child A to the left of child B) or to the parent (aligned to the top of the parent).

Web View

Displays web pages.

Layouts build with adapters:

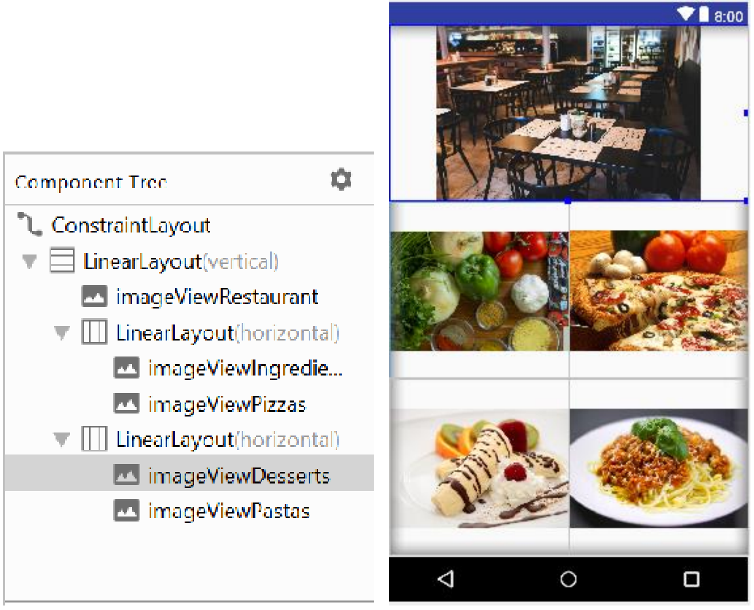

List View
Grid View
Displays a scrolling single column list.
Displays a scrolling grid of columns and rows.

More information on layouts:
https://developer.android.com/guide/topics/ui/declaring-layout

3.2. Linear Layout

Create a new project with an empty activity and name pizzaria. Use the available images in pizzeria.zip (in alternative you may look for images from a restaurant and menus and, if necessary, resize pictures to 720p and save them as .png files), and drag them to the drawable/res directory.

Create the following layout. You might need to use the android:layout_weight parameter to obtain a similar organization space.
https://developer.android.com/guide/topics/ui/layout/linear



```
Component Tree
ConstraintLayout
  LinearLayout(vertical)
    imageViewRestaurant
    LinearLayout(horizontal)
      imageViewIngredie...
      imageViewPizzas
    LinearLayout(horizontal)
      imageViewDesserts
      imageViewPastas
```

3.3. Create a List Activity - Ingredients

A list Adapter allow to associate a DataSource (an array for example) to an Adapter View (ListView, GridView) using a ListAdapter.

Start by creating a new array in the file res>values> string.xml with at list 10 ingredients of your choice.

```
<string-array name="ingredients">
    <item>Ham</item>
    <item>Beef</item>
    <item>Onion</item>
    <item>Pepperoni</item>
    <item>Cheese</item>
    <item>Bacon</item>
    <item>Egg</item>
    <item>Mushroom</item>
</string-array>
```

Add a new empty activity named ingredientActivity. Replace the extend of the activity from AppCompatActivity to ListActivity. Then, modify the code of the new activity according to the following:

```
String [] ingredients;

@Override
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    ingredients = getResources().getStringArray(R.array.ingredients);

    ArrayAdapter<String> adapter = new ArrayAdapter<String>(
            this,
            android.R.layout.simple_list_item_multiple_choice,
            ingredients);

    setListAdapter(adapter);
}

public void onListItemClick (ListView l, View v, int position, long id)
{
    l.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
    l.setItemChecked(position, true);
}
```

Add the method viewIngredients to the main Activity and associate it to the click on the Ingredients Image to open the corresponding activity.

More information: https://developer.android.com/guide/topics/ui/binding

You may try different types of lists suche as:

- simple_list_item1
- simple_list_item_checked
- simple_list_item_multiple_choice)

Add a Toast Message when clicking on an item on the list using the following code:

```
Toast.makeText(this, "add" + l.getItemAtPosition(position).toString(),
        Toast.LENGTH_SHORT).show();
```

3.4. Pizza (GridViews)

Move the images from the pizzas (pizzas.zip) to the drawable directory and add a new empty activity called PizzaActivity.

Use a GridView Layout:

```
<GridView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/pizzaGridView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".PizzaActivity">
</GridView>
```

More information at:
https://developer.android.com/guide/topics/ui/layout/gridview.html

Add the following private array to the class:

```
// Pizza Ids
private Integer [] imageIds = {
        R.drawable.pizza_1, R.drawable.pizza_2,
        R.drawable.pizza_3, R.drawable.pizza_4,
        R.drawable.pizza_5, R.drawable.pizza_6,
        R.drawable.pizza_7, R.drawable.pizza_8,
        R.drawable.pizza_9, R.drawable.pizza_10,
        R.drawable.pizza_11,R.drawable.pizza_12,
        R.drawable.pizza_13,R.drawable.pizza_14,
        R.drawable.pizza_15,R.drawable.pizza_16,
        R.drawable.pizza_17
};

// Pizza names
private String [] pizzaNames = {
        "Pizza Indiana", "Pizza Brunch",
        "Pizza Rainha", "Pizza Burguer",
        "Pizza Barbecue", "Pizza Funny Bacon",
        "Pizza Barbecue-Cream", "Pizza Hot Pepperoni",
        "Pizza Crispy Bacon", "Pizza Camarão",
        "Pizza Tuna", "Pizza Europa",
        "Pizza Americana", "Pizza Especial de Cogumelos",
        "Pizza Especial de Cebola", "Pizza Verdini",
        "Pizza Vegetariana", "Pizza Bananás",
};
```

Modify the OnCreate Method according to the following code:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_pizza);

    GridView gridview = (GridView) findViewById(R.id.pizzaGridView);
    gridview.setAdapter(new ImageAdapter(this));

    // event onClick on each item
    gridview.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        public void onItemClick(AdapterView parent, View v, int position,
long id) {
            Toast.makeText(PizzaActivity.this, pizzaNames[position],
Toast.LENGTH_SHORT).show();
```

```
        }
    });
}
```

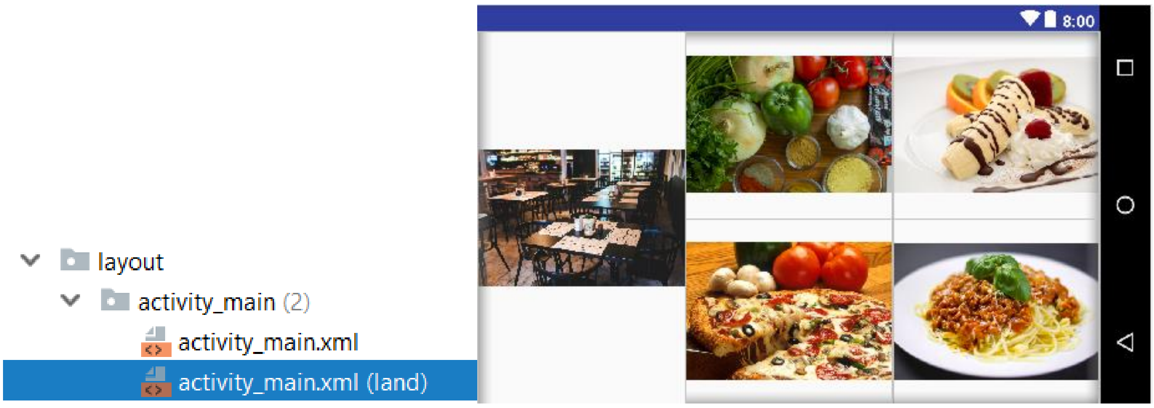
Finally implement the class imageAdapter:

```
public class ImageAdapter extends BaseAdapter {
    private Context mContext;

    public ImageAdapter(Context c) {
        mContext = c;
    }

    public int getCount() {
        return imageIds.length;
    }

    public Object getItem(int position) {
        return null;
    }

    public long getItemId(int position) {
        return 0;
    }

    // Create a new ImageView for each item in the adapter
    public View getView(int position, View convertView, ViewGroup parent) {
        ImageView imageView = new ImageView(mContext);
        imageView.setImageResource(imageIds[position]);
        imageView.setPadding(8, 8, 8, 8);
        return imageView;
    }
}
```

Add the method viewPizzas to the main Activity and associate it to the click on the Pizzas Image to open the corresponding activity.

3.5. Portrait and landscape layout

Create a landscape layout (res>layout>activity_main.xml) and add a new landscape layout. To do so, select the activity_main.xml. Then in design mode choose the option create landscape variation and adjust the layout (switch horizontal to vertical and vice-versa).


layout
activity_main (2)
activity_main.xml
activity_main.xml (land)
8:00